



National Semiconductor
Application Note 485
Ashok Krishnamurthy
April 1987

Digital Filtering Using the HPC

INTRODUCTION

This report discusses the implementation of Infinite Impulse Response (IIR) digital filters using the National Semiconductor HPC microcontroller. A general program, that can be used to implement cascaded second order sections, up to a maximum of 8 sections, is also included. The program may have to be modified for specific A/D and D/A interfaces.

This report is not intended to be a tutorial on Digital Filter Design methods or their implementation details. Such information can be found in references 1 and 2 below. The general discussion included here closely follows that in reference 3.

DIGITAL FILTERING

The general IIR filter with input $x(n)$ and output $y(n)$ can be described by a transfer function of the form

$$H(z) = \frac{Y(z)}{X(z)} = \frac{a(0) + a(1)z^{-1} + \dots + a(m)z^{-m}}{1 + b(1)z^{-1} + \dots + b(p)z^{-p}}$$

To minimize the effects of coefficient truncation, high order filters are usually implemented as a cascade of second order sections. (Another possible choice is parallel realization—see references below).

In cascade realizations, the numerator and denominator polynomials in the above are factored into second order terms, and the filter is realized as a cascade of such second order sections. This is shown in *Figure 1*. A typical second order section has a transfer function of the form

$$H(z) = \frac{A_0 + A_1 z^{-1} + A_2 z^{-2}}{1 + B_1 z^{-1} + B_2 z^{-2}}$$

A second order section such as the above can be realized in a number of ways; the one of concern here is the so-called 1-D form (see Reference 3). The second order 1-D form is shown in *Figure 2*. Based on this figure, we can obtain the following equations:

$$m(k) = x(k) - B_1 \times m(k-1) - B_2 \times m(k-2)$$

$$y(k) = A_0 \times m(k) + A_1 \times m(k-1) + A_2 \times m(k-2)$$

Define $T_1 = -B_1 \times m(k-1) - B_2 \times m(k-2)$,

$$T_2 = A_1 \times m(k-1) + A_2 \times m(k-2)$$



FIGURE 1. Cascade Realization of a Digital Filter

TL/DD/9247-1

FIGURE 2. One Second Order Section

Since T_1 and T_2 depend on signal values at time $k-1$ and $k-2$, we can precompute and store these quantities in the time interval from $k-1$ to k . Then, when $x(k)$ becomes available at time k , $y(k)$ and $m(k)$ can be quickly computed using

$$m(k) = x(k) + T_1,$$

$$y(k) = A_0 \times m(k) + T_2$$

If there are a number of stages, then these computations should be repeated for each stage. Based on these discussions, the operation of a digital filter can be described using the flowchart in *Figure 3*.

USING THE FILTER PROGRAM

Appendix A contains the listing of the program FILTER that can be used to implement cascaded IIR filters as described above. The program as shown uses a codec interfaced to the HPC using MICROWIRE/PLUS™ to do the A/D and D/A conversion. The program can be used with other A/D and D/A converters by suitably modifying the following subroutines: INPUT, OUTPUT and INIT. Only the portions of INIT that deal with the codec interface need to be modified.

TL/DD/9247-1

The filter coefficients and the number of cascaded stages need to be supplied to the program. This is done as follows:

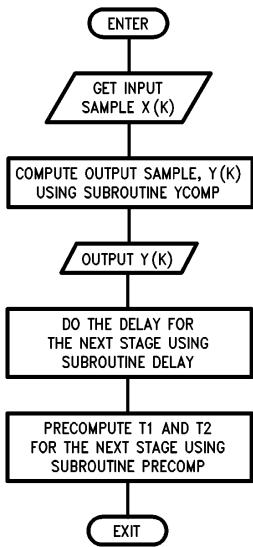
1. *Specification of filter order.* Define a word address called ROMNST and store the number of cascaded stages in that word. The program is presently set up for 4 cascaded stages.

2. *Specification of filter coefficients.* Each second order stage needs the specification of 5 coefficients, A0, A1, A2, B1 and B2. If the number of stages is m, let the coefficients be

A0-1, A1-1, A2-1, B1-1, B2-1 for stage 1,
A0-2, A1-2, A2-2, B1-2, B2-2 for stage 2,

.

A0-m, A1-m, A2-m, B1-m, B2-m for stage m.



TL/DD/9247-3

FIGURE 3. Flowchart for the Computations in a Second Order Module (Based on Reference 3)

Define 5 word addresses called ROMA0, ROMA1, ROMA2, ROMB1, ROMB2 and store these coefficients at these addresses as follows:

ROMA0: .WORD A0-1, A0-2, A0-3, ... A0-m
 ROMA1: .WORD A1-1, A1-2, A1-3, ... A1-m
 ROMA2: .WORD A2-1, A2-2, A2-3, ... A2-m
 ROMB1: .WORD B1-1, B1-2, B1-3, ... B1-m
 ROMB2: .WORD B2-1, B2-2, B2-3, ... B2-m.

Note that the coefficients are signed and need to be in 2's complement representation. Also, the stored coefficients need to be half their actual value. This is because of the way that the program does 2's complement multiplication using the subroutine SMULT.

The FILTER program copies all the coefficients to on-chip RAM for faster execution. Also temporary storage for m (k), m (k - 1), m (k - 2), T1 and T2 is obtained from on-chip RAM. This, along with the storage of various addresses used by the program consumes the entire 192 bytes of user base page RAM.

Note that the filter program does not check for overflow during the various additions. This is because the HPC does not have a signed addition/subtraction overflow flag, and it was felt that the simulation of this feature in software would add excessive overhead. It is therefore the user's responsibility to ensure that the filter coefficients are properly scaled so that the overflow will not occur.

16 x 16 2's COMPLEMENT MULTIPLICATION

One of the basic operations in digital filtering is that of signed multiplication. Since the HPC supports unsigned multiplication only, a method to perform 2's complement multiply using the unsigned multiply is needed.

Let A and B be 2's complement 16 bit integers. Consider the following cases.

1. $A \geq 0, B \geq 0$. In this case the unsigned multiply result is $A \times B$, which is also the 2's complement multiply result. Thus no further processing is needed.
2. $A \geq 0, B < 0$. In this case the unsigned multiply result is $(2^{16}) \times A - A \times |B|$. However the desired result is $(2^{32}) - A \times |B|$. Thus we need to add $(2^{32}) - (2^{16}) \times A$ to the unsigned multiply result to obtain the correct value.
3. $A < 0, B \geq 0$. This case is similar to the previous one. $(2^{32}) - (2^{16}) \times B$ should be added to the unsigned multiply result to get the correct answer.
4. $A < 0, B < 0$. The unsigned multiply result in this case is $(2^{32}) - (2^{16}) \times (|A| + |B|) + |A| \times |B|$. The desired result in this case is $|A| \times |B|$. To get the correct answer, add $(2^{16}) \times (|A| + |B|)$ to the unsigned multiply result.

Based on the above discussion, an algorithm for 2's complement multiplication, where the result is a 32 bit 2's complement integer is shown in *Figure 4*.

1. Let A and B be the two 2's complement integers to be multiplied.
2. Compute $C = A \times B$, the unsigned product of A and B. Let the upper half of C be C-hi and its lower half C-lo.
3. If A is negative, then add $(2^{16}) - B$ to C-hi. This can be easily done using the SET C, SUBC instructions of the HPC. Let the result be C-hi1.
4. If B is negative, then add $(2^{16}) - A$ to C-hi1. Again it is easily done using the SET C, SUBC instructions. Let the result be C-hi2.
5. The 2's complement product of A and B is C-hi2. C-lo.

FIGURE 4. Algorithm for 2's Complement Multiplication.

MULTIPLICATION BY FILTER COEFFICIENTS

The coefficients that arise in most IIR filter designs are numbers that are usually in the range from $-2 < \text{coeff} < 2$. The coefficients, in most instances can be scaled to be in this range. The action of digital filtering involves successive multiplications. If we want no loss in accuracy due to multiplication, the word length needed to store successive partial products increases rapidly—clearly an impractical choice. Thus the results of the multiplication at the various stages need to be truncated to 16 bits before proceeding to the next stage. The program FILTER does this as follows: The filter state variables are regarded as integers, while the filter coefficients are regarded as fixed point fractions with the binary point to the immediate right of the sign bit. After the multiplication, the result is shifted so that the integer part of the product is in one word, and the fractional part in another. The integer part is then returned as the result of the multiplication, i.e. the product is truncated to 16 bits. This is per-

formed by the subroutine SMULT. Since the filter coefficients are regarded as fixed point fractions, only coefficients in the range $-1 < \text{coeff} < 1$ can be represented. However, as discussed earlier, the coefficients are usually in the $-2 < \text{coeff} < 2$ range. This is handled by storing half the coefficient value, and SMULT performs a multiplication by 2 (Shift left) to compensate for it. This is why the coefficient values need to be half their value—a fact mentioned earlier.

REFERENCES

1. A.V. Oppenheim and R.W. Schafer, *Digital Signal Processing*, Prentice-Hall, New Jersey, 1975.
2. L.R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*, Prentice-Hall, New Jersey, 1975.
3. H.T. Nagle and V.P. Nelson, "Digital Filter Implementation on 16-bit Microcomputers", *IEEE Micro*, Feb. 1981, pp. 23-41.

The code listed in this App Note is available on Dial-A-Helper.

Dial-A-Helper is a service provided by the Microcontroller Applications Group. The Dial-A-Helper system provides access to an automated information storage and retrieval system that may be accessed over standard dial-up telephone lines 24 hours a day. The system capabilities include a MESSAGE SECTION (electronic mail) for communicating to and from the Microcontroller Applications Group and a FILE SECTION mode that can be used to search out and retrieve application data about NSC Microcontrollers. The minimum system requirement is a dumb terminal, 300 or 1200 baud modem, and a telephone. With a communications package and a PC, the code detailed in this App Note can be downloaded from the FILE SECTION to disk for later use. The Dial-A-Helper telephone lines are:

Modem (408) 739-1162
Voice (408) 721-5582

For Additional Information, Please Contact Factory

APPENDIX A**Listing of Code for the Program FILTER**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 1

HPC CROSS ASSEMBLER, REV:C, 30 JUL 86

FILTER

```
1      ;
2      ; THIS IS A DEMO PROGRAM TO ILLUSTRATE THE IMPLEMENTATION OF A DIGITAL
3      ; FILTER ON THE HPC. THE PROGRAM CAN BE USED TO IMPLEMENT CASCADED
4      ; SECOND ORDER STAGES. THE MAXIMUM NUMBER OF CASCADED STAGES POSSIBLE
5      ; IS 8 (I.E. THE MAXIMUM FILTER ORDER IS 16).
6      ;
7      ; THE PROGRAM IS DESIGNED FOR THE ANALOG INTERFACE BEING THROUGH
8      ; A CODEC. THE CODEC OUTPUT AND INPUT ARE INTERFACED TO THE HPC USING
9      ; MICROWIRE/PLUS. THIS RESTRICTS THE SAMPLING RATE TO 8 KHZ. ALSO, AT
10     ; THIS SAMPLING RATE, THE HPC CAN ONLY IMPLEMENT A SECOND ORDER FILTER.
11     ; IF A DIFFERENT ANALOG INTERFACE THAT ALLOWS A LOWER SAMPLING RATE IS
12     ; USED, HIGHER ORDER FILTERS CAN BE IMPLEMENTED. THIS WILL INVOLVE CHANGES
13     ; TO THE FOLLOWING SUBROUTINES: INPUT, OUTPUT AND THE PORTIONS OF INIT
14     ; CONCERNED WITH CODEC INITIALIZATION.
15     ;
16     ; THE PROGRAM IS BASED ON THE DESCRIPTION GIVE IN:
17     ;
18     ;      H.T. NAGLE AND V.P. NELSON, "DIGITAL FILTER IMPLEMENTATION
19     ;      ON 16-BIT MICROCOMPUTERS," IEEE MICRO, FEB. 1981, 23-41.
20     ;
21     ;
22          .TITLE FILTER
23     ;
24     ; DEFINE FILTER VARIABLES AND STORAGE.
25     ;
26     0000      YOUT = M(00)           ; OUTPUT SAMPLE STORAGE.
27     0002      YOFK = W(02)           ; TEMPORARY STORAGE.
28     0004      NSTG = W(04)           ; NUMBER OF FILTER STAGES.
29     0006      NCNT = W(06)           ; TEMPORARY STORAGE.
30     0008      PTEMP = W(08)           ; TEMPORARY STORAGE.
31     000A      MTEMP = W(0A)           ; TEMPORARY STORAGE.
32     000C      A0ADDR = W(0C)          ; ADDRESS OF START OF A0 AREA.
33     000E      A1ADDR = W(0E)          ; ADDR. OF START OF A1 AREA.
34     0010      A2ADDR = W(010)         ; ADDR. OF START OF A2 AREA.
35     0012      B1ADDR = W(012)         ; ADDR. OF START OF B1 AREA.
36     0014      B2ADDR = W(014)         ; ADDR. OF START OF B2 AREA.
37     0016      MOADDR = W(016)         ; ADDR. OF START OF MO AREA.
38     0018      M1ADDR = W(018)         ; ADDR. OF START OF M1 AREA.
39     001A      M2ADDR = W(01A)         ; ADDR. OF START OF M2 AREA.
40     001C      T1ADDR = W(01C)         ; ADDR. OF START OF T1 AREA.
41     001E      T2ADDR = W(01E)         ; ADDR. OF START OF T2 AREA.
42     ;
43     0020      A0 = W(020)           ; COEFF. A0.
44     0030      A1 = W(030)           ; COEFF. A1.
45     0040      A2 = W(040)           ; COEFF. A2.
46     0050      B1 = W(050)           ; COEFF. B1.
47     0060      B2 = W(060)           ; COEFF. B2.
48     0070      MO = W(070)           ; M(K).
49     0080      M1 = W(080)           ; M(K-1).
50     0090      M2 = W(090)           ; M(K-2).
51     00A0      T1 = W(0A0)           ; T1.
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 2
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
52      00B0          T2 = W(0B0)           ; T2.  
53      ;  
54      ; DEFINITION OF HPC REGISTER NAMES.  
55      ;  
56      00C0          PSW = M(00C0)  
57      00D0          ENIR = M(00D0)  
58      00D2          IRPD = M(00D2)  
59      00D4          IRCD = M(00D4)  
60      00D6          SIO = M(00D6)  
61      00D8          PORTI = M(00D8)  
62      00E2          PORTBL = M(00E2)  
63      00E3          PORTBH = M(00E3)  
64      00E2          PORTB = W(00E2)  
65      00F2          DIRBL = M(00F2)  
66      00F3          DIRBH = M(00F3)  
67      00F2          DIRB = W(00F2)  
68      00F4          BFUNL = M(00F4)  
69      00F5          BFUNH = M(00F5)  
70      00F4          BFUN = W(00F4)  
71      0188          T2TIM = W(0188)  
72      0186          T2REG = W(0186)  
73      018E          DIVBYL = M(018E)  
74      018F          DIVBYH = M(018F)  
75      018E          DIVBY = W(018E)  
76      0190          TMMDL = M(0190)  
77      0191          TMMDH = M(0191)  
78      0190          TMMMD = W(0190)  
79      ;  
80      ; INCLUDE THE MU-LAW TO LINEAR CODE CONVERSION TABLE.  
81      ;  
82      ;.INCLD MUTBL.MAC  
83 F000          MUTBL, 0F000  
84      ;  
85 F200          . = 0F200  
86      FILTER:  
87 F200 B701FOC4    LD SP, 01FO        ; INITIALIZE STACK POINTER.  
88 F204 305A       JSR INIT         ; INITIALIZE THE CODEC  
89                      ; AND FILTER VARIABLES.  
90      ; NEXT COMES THE BASIC FILTER LOOP.  
91      FLOOP:  
92 F206 3101       JSR INPUT        ; GET INPUT SAMPLE, OUTPUT  
93                      ; PREVIOUS FILTER OUTPUT.  
94 F208 311B       JSR YCOMP        ; COMPUTE NEW OUTPUT.  
95 F20A 315B       JSR OUTPUT        ; CONVERT OUTPUT VALUE TO  
96                      ; MU-255 LAW AND SAVE.  
97 F20C 31AA       JSR PRECOM       ; PRECOMPUTE FOR NEXT SAMPLE.  
98 F20E 68          JP FLOOP        ; GO DO NEXT SAMPLE.  
99      .LOCAL  
100     ;  
101     ;  
102     ;
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 3
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
103      ; THIS SUBROUTINE COMPUTES 2*F*I, WHERE F IS A 2'S COMPLEMENT
104      ; BINARY FRACTION AND I IS A 2'S COMPLEMENT INTEGER. THE INTEGER
105      ; PART OF THE PRODUCT IS RETURNED IN A. ON INPUT, EITHER F OR I
106      ; SHOULD BE IN A AND THE ADDRESS OF THE OTHER IN B.
107      ;
108      ;
109      SMULT:
110     F20F B700000A          LD MTEMP, 0           ; CLEAR TEMPORARY STORAGE.
111     F213 A9CC              INC B                ; B NOW POINTS TO UPPER BYTE
112
113     F215 17                IF M(B).7           ; OF MULTIPLIER.
114     F216 ABOA              ST A, MTEMP         ; IS IT NEGATIVE?
115     F218 AACC              DECSZ B            ; THEN SAVE MULTPLICAND IN MTEMP.
116     F21A 40                NOP                 ; B INTO WORD POINTER.
117     F21B AE0A              X A, MTEMP         ; SWAP A AND MTEMP.
118     F21D 960B17              IF M((\$MTEMP) + 1).7 ; IS MULTPLICAND NEGATIVE?
119     F220 F8                ADD A, W(B)        ; THEN ACCUMULATE MULTIPLIER
120     F221 AE0A              X A, MTEMP         ; UNSIGNED MULTIPLY.
121     F223 FE                MULT A, W(B)       ; UPPER HALF IN A.
122     F224 AECE              X A, X             ; SET C
123     F226 02                SET C
124     F227 960AEB             SUBC A, MTEMP
125     F22A E7                SHL A
126     F22B 96CF17              IF H(X).7
127     F22E 04                INC A
128     F22F E7                SHL A
129     F230 96CF16              IF H(X).6
130     F233 04                INC A
131     F234 3C                RET
132
133
134      ; THIS SUBROUTINE PERFORMS THE INITIALIZATION FOR THE FILTER.
135      ; IT DOES THE FOLLOWING:
136      ;   1. SET UP THE FILTER VARIABLES.
137      ;   2. COPY THE FILTER COEFFS. FROM ROM TO ON CHIP RAM.
138      ;   3. INITIALIZE AND START THE CODEC.
139
140
141      ; DEFINE FILTER COEFFICIENTS.
142
143     F235 40                .EVEN
144     F236 0400              ROMNST:    .WORD 4
145     F238 C430              ROMAO:     .WORD 12484, 3217, 4574, 7636
146
147     F23A 910C              F23C DE11
148     F23E D41D              F242 910C
149     F244 DE11              F246 D41D
150
151     F248 C430              ROMAL:    .WORD 12484, 3217, 4574, 7636
152
153     F24A 910C              ROMA2:    .WORD 12484, 3217, 4574, 7636
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 4
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
F24A 910C
F24C DE11
F24E D41D
148 F250 B939      ROMB1:      .WORD 14777, 9826, 19308, 11207
F252 6226
F254 6C4B
F256 C72B
149 F258 A1D5      ROMB2:      .WORD -10847, -15783, -6940, -14068
F25A 59C2
F25C E4E4
F25E OCC9
150          INIT:
151 F260 B6F236A8    LD A, W(ROMNST)
152 F264 AB04        ST A, NSTG           ; SET UP NO. OF STAGES.
153 F266 9020        LD A, $AO
154 F268 AB0C        ST A, AOADDR        ; COPY ADDRESS OF AO AREA.
155 F26A 9030        LD A, $A1
156 F26C AB0E        ST A, A1ADDR        ; COPY ADDRESS OF A1 AREA.
157 F26E 9040        LD A, $A2
158 F270 AB10        ST A, A2ADDR        ; COPY ADDRESS OF A2 AREA.
159 F272 9050        LD A, $B1
160 F274 AB12        ST A, B1ADDR        ; COPY ADDRESS OF B1 AREA.
161 F276 9060        LD A, $B2
162 F278 AB14        ST A, B2ADDR        ; COPY ADDRESS OF B2 AREA.
163 F27A 9070        LD A, $M0
164 F27C AB16        ST A, MOADDR        ; COPY ADDRESS OF M0 AREA.
165 F27E 9080        LD A, $M1
166 F280 AB18        ST A, MIADDR        ; COPY ADDRESS OF M1 AREA.
167 F282 9090        LD A, $M2
168 F284 AB1A        ST A, M2ADDR        ; COPY ADDRESS OF M2 AREA.
169 F286 90A0        LD A, $T1
170 F288 AB1C        ST A, T1ADDR        ; COPY ADDRESS OF T1 AREA.
171 F28A 9080        LD A, $T2
172 F28C AB1E        ST A, T2ADDR        ; COPY ADDRESS OF T2 AREA.
173          ;
174          ; COPY THE AO COEFFS. TO ON-CHIP RAM.
175          ;
176 F28E B3F238      LD X, ROMAO
177 F291 9220        LD B, $AO
178 F293 AC04CA      LD K, NSTG
179          CAOLP:
180 F296 F0          LD A, W(X+)
181 F297 E1          XS A, W(B+)
182 F298 40          NOP
183 F299 AAC9        DECSZ K
184 F29B 65          JP CAOLP
185          ;
186          ; COPY THE A1 COEFFS. TO ON-CHIP RAM.
187 F29C B3F240      LD X, ROMA1
188 F29F 9230        LD B, $A1
189 F2A1 AC04CA      LD K, NSTG
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 5
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
190          CALLP:  
191 F2A4 F0          LD A, W(X+)  
192 F2A5 E1          XS A, W(B+)  
193 F2A6 40          NOP  
194 F2A7 AAC A       DECSZ K  
195 F2A9 65          JP CALLP  
196          ;  
197          ; COPY THE A2 COEFFS. TO ON-CHIP RAM.  
198 F2AA B3F248      LD X, ROMA2  
199 F2AD 9240        LD B, $A2  
200 F2AF AC04CA      LD K, NSTG  
201          CA2LP:  
202 F2B2 F0          LD A, W(X+)  
203 F2B3 E1          XS A, W(B+)  
204 F2B4 40          NOP  
205 F2B5 AAC A       DECSZ K  
206 F2B7 65          JP CA2LP  
207          ;  
208          ; COPY THE B1 COEFFS. TO ON-CHIP RAM.  
209 F2BB B3F250      LD X, ROMB1  
210 F2BB 9250        LD B, $B1  
211 F2BD AC04CA      LD K, NSTG  
212          CB1LP:  
213 F2C0 F0          LD A, W(X+)  
214 F2C1 E1          XS A, W(B+)  
215 F2C2 40          NOP  
216 F2C3 AAC A       DECSZ K  
217 F2C5 65          JP CB1LP  
218          ;  
219          ; COPY THE B2 COEFFS. TO ON-CHIP RAM.  
220 F2C6 B3F258      LD X, ROMB2  
221 F2C9 9260        LD B, $B2  
222 F2CB AC04CA      LD K, NSTG  
223          CB2LP:  
224 F2CE F0          LD A, W(X+)  
225 F2CF E1          XS A, W(B+)  
226 F2D0 40          NOP  
227 F2D1 AAC A       DECSZ K  
228 F2D3 65          JP CB2LP  
229          ;  
230          ; ZERO OUT THE REST OF USER BASE PAGE RAM.  
231          ;  
232 F2D4 8D70BE      LD BK, $MO, OBE  
233          ZEROLP:  
234 F2D7 00          CLR A  
235 F2D8 E1          XS A, W(B+)  
236 F2D9 62          JP ZEROLP  
237          ;  
238          ;  
239          ; NOW INITIALIZE AND START THE CODEC.  
240          ;
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 6
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
241          ;  
242 F2DA B7FFB7F2      LD DIRB,OFFB7      ; SET B3 (T2IO) AND B6 (SK)  
243          ; ON PORT B AS INPUTS. SET ALL  
244          ; OTHER PINS ON B AS OUTPUT.  
245 F2DE B70000E2      LD PORTB, 0       ; OUTPUT 0 ON ALL PORT B PINS.  
246 F2E2 96F40B        SET BFUNL.3      ; ALT. FUN. ON B3-T2IO.  
247 F2E5 96F40D        SET BFUNL.5      ; ALT. FUN. ON B5-S0.  
248 F2E8 96F508        SET BFUNH.0      ; ALT. FUN. ON B8-TSO.  
249 F2EB 9700D0        LD ENIR, 0       ; DISABLE INTRPTS.  
250 F2EE 9700D4        LD IRCD, 0       ; SELECT SLAVE MODE FOR M-WIRE.  
251 F2F1 83070188AB    LD T2TIM, 07     ; LOAD 7-DEC INTO T2 TIMER.  
252 F2F6 83070186AB    LD T2REG, 07     ; LOAD 7-DEC INTO T2 REG.  
253 F2FB 8300018F8B    LD DIVBYH, 0     ; SELECT EXT. CLOCK FOR T2 TIMER.  
254          ;  
255 F300 8ED6          X A, SIO  
256 F302 8740400190AB   LD TMMD, 04040    ; START TIMER T2.  
257 F308 3C             RET  
258          ;  
259          ;  
260          ; THIS SUBROUTINE OUTPUTS THE PREVIOUS Y(K) TO THE CODEC AND READS  
261          ; THE NEW INPUT VALUE. THEN THE MU-255 VALUE IS CONVERTED TO LINEAR  
262          ; BY TABLE LOOK UP. THE TABLE IS ASSUMED TO START AT F000.  
263          ;  
264          ;  
265 INPUT:  
266 F309 AB02          LD A, YOKF       ; GET DATA TO BE OUTPUT.  
267 NOTDN:  
268 F30B 96D210         IF IRPD.0      ; IS MICROWIRE DONE?  
269 F30E 41             JP MWDONE     ; YES, SO GET DATA.  
270 F30F 64             JP NOTDN      ; NO, SO TRY AGAIN.  
271 MWDONE:  
272 F310 8ED6          X A, SIO       ; GET NEW SAMPLE, OUTPUT  
273          ; COMPUTED DATA.  
274 F312 01             COMP A        ; TAKE CARE OF CODEC INVERSION.  
275 F313 99FF          AND A, OFF  
276 F315 E7             SHL A  
277 F316 BAF000         OR A, OF000    ; FORM MU-LAW TO LINEAR  
278          ; TABLE ADDRESS.  
279 F319 AECE          X A, X  
280 F31B D0             LD A, M(X+)    ; GET LINEAR VALUE  
281 F31C AECA          X A, K  
282 F31E D4             LD A, M(X)    ; A BYTE AT A TIME.  
283 F31F 8CC8CB         LD H(K), L(A)  
284 F322 A8CA          LD A, K  
285 F324 3C             RET  
286          ;  
287          ;  
288 YCOMP:  
289          ; THIS SUBROUTINE COMPUTES THE OUTPUT SAMPLE Y(K).  
290          ; THE INPUT SAMPLE X(K) IS INPUT IN REG. A.  
291          ; THE OUTPUT IS RETURNED IN REG. A.
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 7

HPC CROSS ASSEMBLER, REV:C, 30 JUL 86

FILTER

```
292          ;
293 F325 AC0406      LD NCNT, NSTG      ; COPY THE NUMBER OF STAGES TO
294          ; NCNT.
295          YLOOP:
296 F328 AD1CF8      ADD A, W(T1ADDR)    ; A ≤ X(K) + T1.
297 F32B AD16AB      ST A, W(MOADDR)    ; M(K) ≤ X(K) + T1.
298 F32E ACOCCC     LD B, AOADDR      ; B ≤ ADDR(AO).
299 F331 3522        JSR SMULT       ; A ≤ AO*M(K).
300 F333 AD1EF8      ADD A, W(T2ADDR)    ; A ≤ AO*M(K) + T2.
301          ;
302 F336 AA06        DECSZ NCNT      ; DONE ALL STAGES?
303 F338 941B      R   JMP YMORE      ; NO GO DO SOME MORE.
304          ;
305          ; GET HERE MEANS ALL STAGES DONE.
306 F33A AB02        ST A, YOFK       ; SAVE TEMPORARILY.
307 F33C A804        LD A, NSTG
308 F33E 05          DEC A
309 F33F E7          SHL A
310 F340 01          COMP A
311 F341 04          INC A          ; A ≤ -2*(NSTG-1).
312 F342 A0C81CF8    ADD TIADDR, A    ; RESTORE TIADDR.
313 F346 A0C816F8    ADD MOADDR, A    ; RESTORE MOADDR.
314 F34A A0C80CF8    ADD AOADDR, A    ; RESTORE AOADDR.
315 F34E A0C81EF8    ADD T2ADDR, A    ; RESTORE T2ADDR.
316 F352 A802        LD A, YOFK       ; A ≤ Y(K).
317 F354 3C          RET
318          ;
319          ; PREPARE FOR NEXT STAGE ITERATION.
320          ;
321          YMORE:
322 F355 82021CF8    ADD TIADDR, 02
323 F359 820216F8    ADD MOADDR, 02
324 F35D 82020CF8    ADD AOADDR, 02
325 F361 82021EF8    ADD T2ADDR, 02
326 F365 953D        JMP YLOOP
327          ;
328          ; THIS SUBROUTINE CONVERTS THE 16 BIT OUTPUT VALUE TO
329          ; 8 BIT MU-LAW.
330          ;
331          OUTPUT:
332 F367 96D41F      RESET IRCD.7
333 F36A E7          SHL A          ; SIGN BIT TO C.
334 F36B 06          IFN C          ; IS IT POSITIVE?
335 F36C 45          JP OPOS
336 F36D 96D40F      SET IRCD.7
337 F370 01          COMP A
338 F371 04          INC A          ; NEGATIVE, SO TAKE 2'S
339          ; COMPLEMENT.
340          ; OPOS:
341 F372 B80108      ADD A, 0108    ; ADD BIAS.
342 F375 9107        LD K, 07      ; SET UP COUNTER.
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 8
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
343          ALIGN:  
344 F377 E7      SHL A           ; LOOP AND LOCATE MS 1 BIT.  
345 F378 07      IF C  
346 F379 44      JP ODONE        ; FOUND MS 1 BIT.  
347 F37A AAC A   DECSZ K  
348 F37C 65      JP ALIGN  
349 F37D E7      SHL A           ; HAS TO BE 1 IN C NOW.  
350          ODONE:  
351 F37E AECA    X R, K  
352 F380 E7      SHL A  
353 F381 E7      SHL A  
354 F382 E7      SHL A  
355 F383 E7      SHL A           ; COUNTER VALUE IN BITS 4-6.  
356 F384 AECC    X A, B  
357 F386 00      CLR A  
358 F387 88CB    LD A, H(K)  
359 F389 3B      SWAP A  
360 F38A 990F    AND A, OF  
361 F38C 96CCFA  OR A, B  
362 F38F 96D417  IF IRCD.7  
363 F392 96C80F  SET A.7  
364 F395 01      COMP A  
365 F396 8B00    ST A, YOUT  
366 F398 3C      RET  
367          ;  
368          ; THIS SUBROUTINE UPDATES M(K-1) AND M(K-2) FOR THE NEXT SAMPLE.  
369          ;  
370 F399 AC1ACC  LD B, M2ADDR    ; B ≤ ADDR(M2),  
371 F39C AC04CA  LD K, NSTG       ; K ≤ NSTG.  
372 F39F AC18CE  LD X, M1ADDR    ; X ≤ ADDR(M1).  
373          DLYLP1:  
374 F3A2 F0      LD A, W(X+)  
375 F3A3 E1      XS A, W(B+)  
376 F3A4 40      NOP  
377 F3A5 AAC A   DECSZ K  
378 F3A7 65      JP DLYLP1  
379          ;  
380 F3A8 AC18CC  LD B, M1ADDR    ; B ≤ ADDR(M1),  
381 F3AB AC04CA  LD K, NSTG       ; K ≤ NSTG.  
382 F3AE AC16CE  LD X, MOADDR    ; X ≤ ADDR(M0).  
383          DLYLP2:  
384 F3B1 F0      LD A, W(X+)  
385 F3B2 E1      XS A, W(B+)  
386 F3B3 40      NOP  
387 F3B4 AAC A   DECSZ K  
388 F3B6 65      JP DLYLP2  
389 F3B7 3C      RET  
390          ;  
391          ;  
392          PRECOMP:  
393          ; THIS SUBROUTINE PRECOMPUTES T1 AND T2 BEFORE THE NEXT INPUT
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 9

HPC CROSS ASSEMBLER, REV:C, 30 JUL 86

FILTER

```
394          ; SAMPLE ARRIVES.  
395          ;  
396 F3B8 AC0406      LD NCNT, NSTG      ; COPY NO. OF STAGES.  
397          PRELP:  
398 F3BB AD18A8      LD A, W(M1ADDR)    ; A ≤ M(K-1).  
399 F3BE AC12CC      LD B, B1ADDR      ; B ≤ ADDR(-B1).  
400 F3C1 35B2      JSR SMULT        ; A ≤ -B1*M(K-1).  
401 F3C3 AB08      ST A, PTEMP       ;  
402 F3C5 AD1AA8      LD A, W(M2ADDR)    ; A ≤ M(K-2).  
403 F3C8 AC14CC      LD B, B2ADDR      ; B ≤ ADDR(-B2).  
404 F3CB 35BC      JSR SMULT        ; A ≤ -B2*M(K-2).  
405 F3CD 9608F8      ADD A, PTEMP       ; A ≤ -B1*M(K-1) - B2*M(K-2).  
406 F3D0 AD1CAB      ST A,W(T1ADDR)  
407 F3D3 AD18A8      LD A, W(M1ADDR)    ; A ≤ M(K-1).  
408 F3D6 ACOECC      LD B, A1ADDR      ; B ≤ ADDR(A1).  
409 F3D9 35CA      JSR SMULT        ; A ≤ A1*M(K-1).  
410 F3DB AB08      ST A, PTEMP       ;  
411 F3DD AD1AA8      LD A, W(M2ADDR)    ; A ≤ M(K-2).  
412 F3E0 AC10CC      LD B, A2ADDR      ; B ≤ ADDR(A2).  
413 F3E3 3504      JSR SMULT        ; A ≤ A2*M(K-2).  
414 F3E5 9608F8      ADD A, PTEMP       ; A ≤ A1*M(K-1) + A2*M(K-2).  
415          ;  
416 F3E8 AA06      DECSZ NCNT       ; DONE ALL STAGES?  
417 F3EA 9427      JMP PMORE       ; NO, GO DO SOME MORE.  
418          ;  
419          ; GET HERE MEANS DONE ALL STAGES.  
420 F3EC A804      LD A, NSTG  
421 F3EE 05      DEC A  
422 F3EF E7      SHL A  
423 F3F0 01      COMP A  
424 F3F1 04      INC A      ; A ≤ -2*(NSTG - 1).  
425 F3F2 A0C818F8      ADD M1ADDR, A    ; RESTORE M1ADDR.  
426 F3F6 A0C81AF8      ADD M2ADDR, A    ; RESTORE M2ADDR.  
427 F3FA A0C81CF8      ADD T1ADDR, A    ; RESTORE T1ADDR.  
428 F3FE A0C81EF8      ADD T2ADDR, A    ; RESTORE T2ADDR.  
429 F402 A0C812F8      ADD B1ADDR, A    ; RESTORE B1ADDR.  
430 F406 A0C814F8      ADD B2ADDR, A    ; RESTORE B2ADDR.  
431 F40A A0C80EF8      ADD A1ADDR, A    ; RESTORE A1ADDR.  
432 F40E A0C810F8      ADD A2ADDR, A    ; RESTORE A2ADDR.  
433 F412 3C          RET  
434          ;  
435          ; PREPARE FOR NEXT STAGE ITERATION.  
436          ;  
437          PMORE:  
438 F413 820218F8      ADD M1ADDR, 02    ; UPDATE M1ADDR.  
439 F417 82021AF8      ADD M2ADDR, 02    ; UPDATE M2ADDR.  
440 F41B 82021CF8      ADD T1ADDR, 02    ; UPDATE T1ADDR.  
441 F41F 82021EF8      ADD T2ADDR, 02    ; UPDATE T2ADDR.  
442 F423 820212F8      ADD B1ADDR, 02    ; UPDATE B1ADDR.  
443 F427 820214F8      ADD B2ADDR, 02    ; UPDATE B2ADDR.  
444 F42B 82020EF8      ADD A1ADDR, 02    ; UPDATE A1ADDR.
```

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 10
HPC CROSS ASSEMBLER, REV:C, 30 JUL 86
FILTER

```
445 F42F 820210F8      ADD A2ADDR, 02      ; UPDATE A2ADDR.  
446 F433 9578          JMP PRELP  
447                 ;  
448                 ;  
449 FFFE 00F2          .END FILTER
```

APPENDIX A (Continued)**Listing of Code for the Program FILTER (Continued)**

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 11

HPC CROSS ASSEMBLER, REV:C, 30 JUL 86

FILTER

SYMBOL TABLE

A 00C8 W	A0 0020 W	AOADDR 000C W	A1 0030 W
A1ADDR 000E W	A2 0040 W	A2ADDR 0010 W	ALIGN F377
B 00CC W	B1 0050 W	B1ADDR 0012 W	B2 0060 W
B2ADDR 0014 W	BFUN 00F4 W*	BFUNH 00F5 M	BFUNL 00F4 M
CA0LP F296	CALLP F2A4	CA2LP F2B2	CB1LP F2C0
CB2LP F2CE	DIRB 00F2 W	DIRBH 00F3 M*	DIRBL 00F2 M*
DIVBY 018E W*	DIVBYH 018F M	DIVBYL 018E M*	DLYLP1 F3A2
DLYLP2 F3B1	ENIR 00D0 M	FILTER F200	FLOOP F206
INCRM 0200	INIT F260	INPUT F309	IRCD 00D4 M
IRPD 00D2 M	K 00CA W	M0 0070 W	MOADDR 0016 W
M1 0080 W	M1ADDR 0018 W	M2 0090 W	M2ADDR 001A W
MTEMP 000A W	MUAL 205F	MWDONE F310	NCNT 0006W
NOTDN F30B	NSTG 0004 W	ODONE F37E	OPOS F372
OUTPUT F367	PC 00C6 W	PMORE F413	PORTB 00E2 W
PORTBH 00E3 M*	PORTBL 00E2 M*	PORTI 0008 M*	PRECOM F3B8
PRELFP F3BB	PSW 00C0 M*	PTEMP 0008 W	ROMAO F238
ROMA1 F240	ROMA2 F248	ROMB1 F250	ROMB2 F258
ROMNST F236	RVAL E0A1	S10 00D6 M	SMULT F20F
SP 00C4 W	SVAL 2100	T1 00A0 W	T1ADDR 001C W
T2 00B0 W	T2ADDR 001E W	T2REG 0186 W	T2TIM 0188 W
TMMD 0190 W	TMMDH 0191 M*	TMMDL 0190 M*	X 00CE W
YCOMP F325	YLOOP F328	YMORE F355	YOFK 0002 W
YOUT 0000 M	ZEROLP F2D7		

APPENDIX A (Continued)

Listing of Code for the Program FILTER (Continued)

NATIONAL SEMICONDUCTOR CORPORATION PAGE: 12

HPC CROSS ASSEMBLER, REV:C, 30 JUL 86

FILTER

MACRO TABLE

MUTBL

NO WARNING LINES

NO ERROR LINES

1079 ROM BYTES USED

SOURCE CHECKSUM = 4769

OBJECT CHECKSUM = 1378

INPUT FILE C:FILTER.MAC

LISTING FILE C:FILTER.PRN

OBJECT FILE C:FILTER.LM

LIFE SUPPORT POLICY

NATIONAL'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF NATIONAL SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and whose failure to perform, when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in a significant injury to the user.
2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

 **National Semiconductor Corporation**
1111 West Bardin Road
Arlington, TX 76017
Tel: (800) 272-9959
Fax: (800) 737-7018

National Semiconductor Europe
Fax: (+49) 0-180-530 85 86
Email: cnjwge@tevm2.nsc.com
Deutsch Tel: (+49) 0-180-530 85 85
English Tel: (+49) 0-180-532 78 32
Français Tel: (+49) 0-180-532 93 58
Italiano Tel: (+49) 0-180-534 16 80

National Semiconductor Hong Kong Ltd.
13th Floor, Straight Block,
Ocean Centre, 5 Canton Rd.
Tsimshatsui, Kowloon
Hong Kong
Tel: (852) 2737-1600
Fax: (852) 2736-9960

National Semiconductor Japan Ltd.
Tel: 81-043-299-2309
Fax: 81-043-299-2408